UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/789,135	02/27/2004	Daryl B. Olander	ORACL-01401US0	9244

23910 7590 08/21/2008

FLIESLER MEYER LLP
650 CALIFORNIA STREET
14TH FLOOR
SAN FRANCISCO, CA 94108

EXAMINER	
HEFFINGTON, JOHN M	
ART UNIT	**PAPER NUMBER**
2179	

MAIL DATE	DELIVERY MODE
08/21/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/789,135	OLANDER ET AL.	
	Examiner	Art Unit	
	JOHN M. HEFFINGTON	2179	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 May 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-40* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-40* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *5/22/08, 8/11/08*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

This action is in response to amended filing of 22 May 2008. Claims 1, 14, 23, 27 and 40 have been amended. Claims 1-40 are pending and have been considered below.

Response to Arguments

1. Applicant's arguments with respect to claims 1, 14, 27 and 40 have been considered but are moot in view of the new ground(s) of rejection.

2. Applicant's arguments filed 24 April 2008 have been fully considered but they are not persuasive.

Moss discloses a technique called HTTP tunneling, wherein an applet executing on a client establishes an HTTP connection with a servlet container and makes method calls on a servlet executing in the servlet container via the HTTP connection. The methods invoked as a result of an event being raised in the applet may reside in the servlet. The examiner has asserted that a servlet is analogous to a control tree since a servlet is generated by a servlet container, analogous to the control tree factory, and a servlet can construct a graphical user interface (GUI). It is important to note that an applet can also be considered to be a control tree, given that an applet can render a GUI with controls which can be represented by a tree, i.e. the controls can be represented hierarchically. An applet contains actual graphical controls, whereas, a servlet represents graphical controls via HTML code which is rendered as a GUI by a browser. A servlet container does not explicitly construct an applet, a servlet merely includes the applet tag in the

HTML code sent to the browser in response to an HTTP request received by the servlet container, and then, when the browser executes the HTML code of the response, the applet is loaded by the browser on the client and executed. So, as can be seen, a control tree can be represented by a servlet, which executes on a server within a servlet container, instantiated when a JSP page is loaded by the servlet container, representing the control tree factory, and an applet, whose tag is included in the HTML code a JSP page. In the case of an applet, the servlet instantiated from the JSP page executes on the server and the applet executes on the client. In both cases of a servlet and an applet, no specific GUI controls reside in a tree structure on the server. The servlet can represent a tree structure on the server, however, the servlet does not contain any explicit GUI controls. The servlet created the GUI controls which are rendered on the client. The applet contains GUI controls which can be represented by a tree structure, but the applet resides on the client.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-6, 27-31 and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Moss (Java Servlets) in view of Houglind et al. (Core JSP) and further in view of Schildt (Java 2).

Claim 1, 14, 27 and 40: Moss discloses a method and computer readable medium for accepting a request comprising:

a. mapping the request to a control tree factory (page 2-3, Figure 1.1)
b. generating a response wherein the response can be used to render at least a portion of a graphical user interface (GUI) (page 2-3, Figure 1.1) [In the art, an HTTP response is used to send HTML to a web browser to build a GUI.]
c. wherein the at least one control can represent a graphical element of a GUI (page 36, Servlet Examble: Properties) [A servlet returns an HTML page back to the client. HTML pages contain graphical elements.]
d. wherein the controls of the control tree intercommunicate by raising events in a raise events lifecycle stage; and wherein the raise events lifecycle stage occurs before a render lifecycle stage (pages 154-158, The EchoSurvey Servlet)[When an HTTP request is received, the parameters of the control objects can be read via method calls to .getParameter() and getParameterValues(). The parameters of the control objects can then be used to affect the values of other control objects that will become part of the HTTP response object. This can be considered to be part of the events stage of the service lifecycle. The render stage occurs when PrintWriter object is flushed]
e. wherein a raise event method is called to raise events in the raise events lifecycle stage (chapter 10: Applet to Servlet Communications, pages 212-214, 228-239, figures 10.9-10.14) [Moss discloses a technique called HTTP tunneling,

wherein an applet executing on a client establishes an HTTP connection with a servlet container and makes method calls on a servlet executing in the servlet container via the HTTP connection. The methods invoked as a result of an event being raised in the applet may reside in the servlet. The examiner has asserted that a servlet is analogous to a control tree since a servlet is generated by a servlet container, analogous to the control tree factory, and a servlet can construct a graphical user interface (GUI). It is important to note that an applet can also be considered to be a control tree, given that an applet can render a GUI with controls which can be represented by a tree, i.e. the controls can be represented hierarchically. An applet contains actual graphical controls, whereas, a servlet represents graphical controls via HTML code which is rendered as a GUI by a browser. A servlet container does not explicitly construct an applet, a servlet merely includes the applet tag in the HTML code sent to the browser in response to an HTTP request received by the servlet container, and then, when the browser executes the HTML code of the response, the applet is loaded by the browser on the client and executed. So, as can be seen, a control tree can be represented by a servlet, which executes on a server within a servlet container, instantiated when a JSP page is loaded by the servlet container, representing the control tree factory, and an applet, whose tag is included in the HTML code a JSP page. In the case of an applet, the servlet instantiated from the JSP page executes on the server and the applet executes on the client. In both cases of a servlet and an applet, no specific GUI controls reside in a tree

structure on the server. The servlet can represent a tree structure on the server, however, the servlet does not contain any explicit GUI controls. The servlet created the GUI controls which are rendered on the client. The applet contains GUI controls which can be represented by a tree structure, but the applet resides on the client.]

but does not disclose

a. generating a control tree from the factory based on the request wherein the control tree can include at least one control
b. advancing the control tree through at least one lifecycle stage based on the request

Hougland discloses generating a control tree from the factory based on the request wherein the control tree can include at least one control (page 6-7, Java Servlets, JavaServer Pages) [JSP is a technology using server-side scripting that is actually translated into Servlets…]. Therefore it would have been obvious to one having ordinary skill in the art at the time of the invention for Moss to add generating a control tree from the factory based on the request wherein the control tree can include at least one control. One would have been motivated to add generating a control tree from the factory based on the request wherein the control tree can include at least one control to Moss because it is a standard use of Servlets to translate a JSP into a Servlet.

Further, Schildt discloses advancing the control tree through at least one lifecycle stage based on the request (page 951, The Life Cycle of a Servlet). Therefore it would have been obvious to one having ordinary skill in the art at the time of the invention for Moss to add advancing the control tree through at least one lifecycle stage based on the request. One would have been motivated to add advancing the control tree through at least one lifecycle stage based on the request to Moss because it is the standard behavior for a Servlet to cycle through a life cycle.

Claims 2, 15 and 28: Moss, Hougland and Schildt disclose the method of claims 1, 14 and 27 and Hougland further discloses creating a metadata representation of a control tree (page 6-7, Java Servlets, JavaServer Pages) [JSP is a technology using server-side scripting that is actually translated into Servlets…] and constructing the control tree based on the metadata representation (page 6-7, Java Servlets, JavaServer Pages) [JSP is a technology using server-side scripting that is actually translated into Servlets…]. Therefore it would have been obvious to one having ordinary skill in the art at the time of the invention for Moss to add creating a metadata representation of a control tree and constructing the control tree based on the metadata representation. One would have been motivated to add creating a metadata representation of a control tree and constructing the control tree based on the metadata representation to moss because Converting JSP to Servlets is a common use of Servlets.

Claims 3 and 29: Moss, Hougland and Schildt disclose the method of claim 1 and 27 and and Moss further discloses wherein: the request one of: an hypertext transfer protocol request (HTTP), simple mail transfer protocol request, an instant messaging request, a request based on a standard protocol; and a request based on a proprietary protocol; and the request originates from one of: a web browser, a instant messaging window, and a process (page 2-3, Figure 1.1).

Claims 4, 17 and 30: Moss, Hougland and Schildt disclose the method of claims 1, 14 and 27, and Moss further discloses: providing the response to a web browser (page 2-3, Figure 1.1).

Claims 5, 18 and 31: Moss, Hougland and Schildt disclose the method of claims 1, 14 and 27 and Moss further discloses wherein: the control tree is driven through the at least one lifecycle stage by an interchangeable lifecycle component (page 2, What are Servlets) [A Servlet is executed by a Web server or Servlet container which is interchangeable].

Claims 6, 19 and 32: Moss, Hougland and Schildt disclose the method of claims 1, 14 and 27 and Hougland further discloses the at least one control has an interchangeable persistence mechanism (page 73-74, Multi-threading and Persistance) [The Servlet is persisted by the Servlet container and the Servlet container is interchangeable.]. Therefore it would have been obvious to one having ordinary skill in the art at the time

of the invention for Moss to add the at least one control has an interchangeable persistence mechanism. One would have been motivated to add the at least one control has an interchangeable persistence mechanism to moss because it is inherent in the use of Servlets for Servlets to be persisted by a container.

Claims 8, 21 and 34: Moss, Hougland and Schildt disclose the method of claims 1, 14 and 27 and Schildt further discloses one of the at least one controls can interact with another of the at least one controls (page 138, Introducing Methods) [It is common in the Java® programming language for classes, Servlets in this case, to interact with other classes, or even with themselves, through methods.] Therefore it would have been obvious to one having ordinary skill in the art at the time of the invention for Moss to add one of the at least one controls can interact with another of the at least one controls. One would have been motivated to add one of the at least one controls can interact with another of the at least one controls to Moss because it is common in the Java® programming language for classes, Servlets in this case, to interact with other classes, or even with themselves, through methods.

Claims 9, 22 and 35: Moss, Hougland and Schildt disclose the method of claim 1 and Hougland further discloses one of the at least one controls can advance through the at least one lifecycle stage in parallel with another of the at least one controls (page 6, Java Servlets) [Each individual Servlet runs as a thread inside the Web server process.]. Therefore it would have been obvious to one having ordinary skill in the art at

the time of the invention for Moss to add one of the at least one controls can advance through the at least one lifecycle stage in parallel with another of the at least one controls. One would have been motivated to add one of the at least one controls can advance through the at least one lifecycle stage in parallel with another of the at least one controls to Moss because it is common for Servlets to run as parallel threads in Servlet container process.

Claims 10, 23 and 36: Moss, Hougland and Schildt disclose the method of claims 1, 14 and 27 and Schildt further discloses

a. the lifecycle stage is one of : init, load state, create child controls, load, raise events, pre-render, render, save state, unload and dispose (page 951, Lifecycle of a Servlet), and
b. wherein the lifecycle stage is part of a dynamically configurable lifecycle (page 951, Lifecycle of a Servlet).

Claims 11, 24 and 37: Moss, Hougland and Schildt disclose the method of claim 1, 14 and 27 and Moss further disclose wherein: the response is one of: an hypertext transfer protocol (HTTP) response, a simple mail transfer protocol response, an instant messaging response, a response based on a standard protocol, and a response based on a proprietary protocol (page 3, figure 1.1).

Claims 12, 25 and 38: Moss, Hougland and Schildt disclose the method of claims 1, 14 and 27 and Schildt further discloses controls can raise events and respond to events (page 654, The Delegation Model, Page 896, Design Patterns for Events) [Controls can be represented by classes of any object-oriented programming language such as Java®. Java® classes can raise and respond to events.]. Therefore it would have been obvious to one having ordinary skill in the art at the time of the invention for Moss to add controls can raise events and respond to events. One would have been motivated to add controls can raise events and respond to events to Moss because it is common for Servlets, which are Java®, to raise and respond to events.

Claims 13, 26 and 39: Moss, Hougland and Schildt disclose the method of claims 1, 14 and 27 above and Schildt further discloses the at least one control can be one of: Book, Page, Window, Menu, Layout, Portlet, Theme, Placeholder, Shell, LookAndFeel, Desktop, Body, Footer, Header, Head, Titlebar, ToggleButton, TreeView, TreeViewWithRadioButtons, TextBox, TextArea, Label, Button and Anchor (Page 886, What is a Java Bean) [A Bean may be visible to an end user. One example of this is a button on a graphical user interface.]. Therefore it would have been obvious to one having ordinary skill in the art at the time of the invention for Moss to add the at least one control can be one of: Book, Page, Window, Menu, Layout, Portlet, Theme, Placeholder, Shell, LookAndFeel, Desktop, Body, Footer, Header, Head, Titlebar, ToggleButton, TreeView, TreeViewWithRadioButtons, TextBox, TextArea, Label, Button and Anchor. One would have been motivated to add the at least one control can be one

of: Book, Page, Window, Menu, Layout, Portlet, Theme, Placeholder, Shell, LookAndFeel, Desktop, Body, Footer, Header, Head, Titlebar, ToggleButton, TreeView, TreeViewWithRadioButtons, TextBox, TextArea, Label, Button and Anchor to Moss in order to create a GUI or some other functionality provided by Java® Beans.

5. Claims 7, 20 and 33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Moss (Java Servlets) in view of Hougland (Core JSP) and Schildt (Java 2) as applied to claims 1, 14 and 27 above, and further in view of Geary (Graphic Java).

Claims 7, 20 and 33: Moss, Hougland and Schildt disclose the method and computer readable medium of claim 1, 14 and 27 but do not disclose wherein: the at least one control can render itself according to a theme. Geary discloses the at least one control can render itself according to a theme (page 317, Pluggable Look and Feel)[Controls can be implemented as one or more classes in an object-oriented programming paradigm, such as Java®. Pluggable Look and Feel can be implemented by Java® GUI classes.]. Therefore, it would have been obvious to one having ordinary skill in the art at the time of the invention for Moss to add the at least one control can render itself according to a theme. It would have been obvious to add the at least one control can render itself according to a theme to Moss in order to be able to have variable appearance of the rendered GUI components.

Conclusion

6. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to John M. Heffington whose telephone number is (571) 270-1696. The examiner can normally be reached on Mon - Fri 8:00 - 5:30 EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Weilun Lo can be reached on (571) 272-4847. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JMH
8/14/08

/Ba Huynh/
Primary Examiner, Art Unit 2179